The relational processing limits of classic and contemporary neural network models of language processing

RESEARCH ARTICLE

The relational processing limits of classic and contemporary neural network models of language processing

Guillermo Puebla[a, b], Andrea E. Martin[c, d, e] and Leonidas A. A. Doumas[a]

[a]Department of Psychology, School of Philosophy, Psychology, and Language Sciences, University of Edinburgh, Edinburgh, United Kingdom; [b]Department of Psychology, Universidad de Tarapacá, Arica, Chile; [c]Language and Computation in Neural Systems Group, Max Planck Institute for Psycholinguistics, Nijmegen, The Netherlands; [d]Psychology of Language Department, Max Planck Institute for Psycholinguistics, Nijmegen, The Netherlands; [e]Donders Centre for Cognitive Neuroimaging, Radboud University, Nijmegen, The Netherlands

ARTICLE HISTORY
Compiled September 3, 2020

ABSTRACT
Whether neural networks can capture relational knowledge is a matter of long-standing controversy. Recently, some researchers have argued that (1) classic connectionist models can handle relational structure and (2) the success of deep learning approaches to natural language processing suggests that structured representations are unnecessary to model human language. We tested the Story Gestalt model, a classic connectionist model of text comprehension, and a Sequence-to-Sequence with Attention model, a modern deep learning architecture for natural language processing. Both models were trained to answer questions about stories based on abstract thematic roles. Two simulations varied the statistical structure of new stories while keeping their relational structure intact. The performance of each model fell below chance at least under one manipulation. We argue that both models fail our tests because they can't perform dynamic binding. These results cast doubts on the suitability of traditional neural networks for explaining relational reasoning and language processing phenomena.

KEYWORDS
Relational reasoning; generalization; language processing; neural networks; deep learning

1. Introduction

The ability to represent and reason in terms of the relations between objects plays a crucial role across many aspects of human cognition, from visual perception (Biederman, 1987), to higher cognitive processes such as analogy (Holyoak, 2012), categorization (Medin, Goldstone, & Gentner, 1993), concept learning (Doumas & Hummel, 2013), and language (Gentner, 2016). Furthermore, comparative evidence suggests that relational thinking may be the key cognitive process distinguishing the abilities of humans from those of other species (Christie & Gentner, 2010; Penn, Holyoak, & Povinelli, 2008). Given the relevance of the capacity to represent and reason about

CONTACT Guillermo Puebla. Email: guillermo.puebla@ed.ac.uk

relations across cognitive domains, several computational models in cognitive science have sought to capture its main characteristics and development (e.g., Chen, Lu, & Holyoak, 2017; Doumas, Hummel, & Sandhofer, 2008; Falkenhainer, Forbus, & Gentner, 1989; Halford, Wilson, & Phillips, 1998; Hummel & Holyoak, 1997, 2003; Kollias & McClelland, 2013; Leech, Mareschal, & Cooper, 2008; Lu, Chen, & Holyoak, 2012; Lu, Wu, & Holyoak, 2019; Van der Velde & De Kamps, 2006).

Computational models of relational thinking differ in their representational assumptions. In the canonical view, relational thinking entails using predicate representations. A predicate is an abstract structure that can be dynamically bound to an argument, specifying a set of properties about that argument (Doumas & Hummel, 2005). For example, *predator*($\mathbf{x}$) specifies a series of properties about the variable $\mathbf{x}$ (e.g., carnivore, hunts, etc.). Predicate representations have two main attributes. In the first place, predicates maintain role-filler independence in that at least some aspect of the semantic content of the predicate is invariant with respect to its arguments. For example, *predator*(fox) and *predator*(lynx) will specify the same set of properties (e.g., carnivore, hunts, etc.) about the objects fox and lynx. In the second place, predicates can be dynamically bound to arguments, namely, fillers can be assigned and reassigned to different roles as needed during processing. That predicates can be dynamically bound to arguments allows a given concept to play different roles at different times or in different situations. For example, in a scene where a fox is preying on a hen, but then a lynx comes and eats the fox, the initial binding of fox to *predator* (i.e., predator(fox)) is easily broken and new binding of fox to *prey* (i.e., prey(fox)) is easily formed. Models based on predicates or formally equivalent systems (i.e., systems that perform dynamic binding of independent representations of roles and fillers, or symbolic systems) successfully account for a wide variety of phenomena in the relational thinking literature (for a review see Forbus, Liang, & Rabkina, 2017).

By contrast, traditional Parallel Distributed Processing (PDP) models explicitly eschew the need for structured representations (see, e.g., Rogers & McClelland, 2014). Representations in a PDP model correspond to patterns of activation across a fixed-size layer of units (i.e., an activation vector). These representations are unstructured in the sense that relational roles and objects are not independently represented, but instead are represented simultaneously as a single entity. PDP approaches to relational thinking seek to obtain relational behavior without invoking symbolic machinery (Kollias & McClelland, 2013; Leech et al., 2008; St. John, 1992; St. John & McClelland, 1990; Yuan, 2017). The reasoning behind these models is that if a traditional PDP model successfully performs some relational reasoning task, then predicates are not strictly necessary for that task, and, by extension, might not actually be accurate approximations of human mental representations. Recently, some researchers have argued that PDP models are capable of handling relational knowledge. Particularly, Rogers and McClelland (2008, 2014) have proposed that the gestalt models of text comprehension (Rabovsky, Hansen, & McClelland, 2018; Rabovsky & McClelland, 2020; Rohde, 2002; St. John, 1992; St. John & McClelland, 1990) exhibit successful effective role-to-filler binding. The evidence presented by these models consist invariably on demonstrations of generalizations to "unseen" sentences. However, as is going to be clear in the simulations of the present work, these "unseen" sentences consist typically of known combinations of roles and concept fillers, which allows these models to succeed in the generalization tests by memorizing combinations of roles and fillers in the training dataset. As to which specific mechanism would allow these models to learn to form role-filler bindings, these researchers usually appeal to the concept of emergence, arguing that domain general learning algorithms such as back-propagation in conjunction

Table 1. Restaurant Script.

Script
1. <agent-1> and <agent-2> decided restaurant 2. Restaurant quality <expensive/cheap> 3. Distance to restaurant <far/near> 4. <agent-1/agent-2> ordered <cheap-wine/expensive-wine> 5. <agent-1/agent-2> paid bill 6. <agent-1/agent-2> tipped waiter <big/small/not> 7. Waiter gave change to <agent-1/agent-2>
Concept restrictions
The roles agent-1 and agent-2 are never 'Lois' or 'Albert'
Deterministic rule
The quality of the restaurant determines the distance completely: $expensive \rightarrow far$, $cheap \rightarrow near$

with the distributed nature of the internal representations of PDP models allows for learning open-ended relations (Rogers & McClelland, 2014).

Some of the optimism in the connectionist literature is based, at least partially, on the achievements of deep learning architectures in natural language processing. For example, Rabovsky et al. (2018) argue that the success of Google's neural machine translation system (Wu et al., 2016) implies that structured representations are, in fact, an obstacle to accurately capturing the subtle regularities of human language (also see Rabovsky & McClelland, 2020). In the present study, we tested the Story Gestalt (SG) model (St. John, 1992) and a Sequence-to-Sequence with Attention (Seq2Seq+Attention) model (Bahdanau, Cho, & Bengio, 2015)—the architecture behind Google's neural machine translation system—in a series of tasks requiring binding a number of concepts to several roles in a story. All stories had relational structure in the sense that (1) the thematic roles were organized in specific ways and (2) filling the roles with different concepts yielded different instantiations of the story. In our simulations we trained both models in a large number of these stories to answer questions about the stories and then tested the models with new (untrained) stories. Importantly, we maintained the relational structure of the test stories relative to the training stories while varying their statistical structure (by changing the stories' typical role fillers) in several ways. Next, we describe the generalities of our task and each model's operation in detail.

Our task, based on the original materials of St. John (1992), consists on answering questions about stories generated by a series of (5) scripts. All the scripts describe events as a sequence of propositions where several concepts play different thematic roles: agent-1, agent-2, topic, patient-theme, recipient-destination, location, manner and attribute. As an illustrative example, consider the Restaurant script (Table 1). This script describes an event where two people go to a restaurant. Each sentence of the Restaurant script defines fillers for some roles. To generate a specific instance of a Restaurant script (i.e., a Restaurant story) the roles are given values corresponding to specific concepts. Table 2 (column 1) presents an example of an instantiated Restaurant story in a pseudo-natural language format. The first sentence of this story corresponds to the proposition: agent-1 = Anne, agent-2 = Gary, topic = decided-to-go, patient-theme = None, recipient-destination = restaurant, location = None, manner = None, attribute = None. Appendix A presents all possible concepts values for each role. Note that our scripts produce stories with no repeated topic concepts across propositions.

Each script implements a tree structure where each node represents a proposition and each branch of the tree represents a story. The scripts also implement rules that

Table 2. Example of a Baseline Story (Restaurant).

Story	Questions	Criteria
1. <Anne> and <Gary> decided restaurant	decided	<Anne> and <Gary> decided restaurant
2. Restaurant quality <expensive>	quality	Restaurant quality <expensive>
3. Distance to restaurant <far>	distance	Distance to restaurant <far>
4. <Anne> ordered <cheap-wine>	ordered	<Anne> ordered <cheap-wine>
5. <Anne> paid bill	paid	<Anne> paid bill
6. <Anne> tipped waiter <big>	tipped	<Anne> tipped waiter <big>
7. Waiter gave change to <Anne>	gave	Waiter gave change to <Anne>

specify the probability of transitioning from one node to another conditioned on the value of a character or location role. For example, a rule in the Restaurant script (see Table 1) specifies that if the restaurant is expensive, it will be located far away.

We trained the models in two different conditions. In the *concept restricted condition*, some character or object names were never used in specific scripts. For example, in the Restaurant stories the characters Lois and Albert were never used to fill the roles agent-1 or agent-2 (see Table 1; Appendix B presents detailed descriptions of the remaining scripts, their concept restrictions and rules). In the *concept unrestricted condition* all concepts were used in all stories. Stories in both conditions were generated according to the following procedure: (1) a script is chosen at random, (2) a sequence of propositions is generated by traversing the probabilistic tree structure of a script and (3) character and vehicles names are given specific values (respecting the script's deterministic rule and the script's concept restrictions in the concept restricted condition).

To get a criterion for each model's performance we designed a *baseline test*. In this test we presented the models trained in the unrestricted condition with concept unrestricted stories and asked questions about the stories. The questions corresponded to the concepts filling the topic role. The models generated an answer in the form of a full proposition. The correct answer was the full proposition in which the topic concept was involved. For example, if a proposition in a restaurant story stated that the "waiter gave change to Anne" and the model was asked about the "gave" proposition the correct answer was "waiter gave change to Anne". Because in our stories there was no repeated topics the correct answer was unequivocal. Table 2 presents an example of a Restaurant baseline story, its questions and their corresponding correct answers.

2. Models

2.1. Story Gestalt model

The SG model (St. John, 1992, see Figure 1) integrates a sequence of propositions into a distributed representation of a story, which is then used to answer questions about the story. The model represents all propositions in its input layer through 137 localist units coding for each possible filler of each role (e.g., there is a unit coding for Albert-agent and another unit coding for Albert-recipient). To represent a complete proposition, the units coding for the concept filling each role are activated. For example, a representation of the sentence "Anne and Gary decided to go to the restaurant" would consist of a vector of 137 units were the three units coding for Anne-agent, Gary-agent, decided-topic and restaurant-location are set to 1 and all other units are set to 0 (Figure 1A).

Figure 1B illustrates the SG model's architecture. The model is composed of two

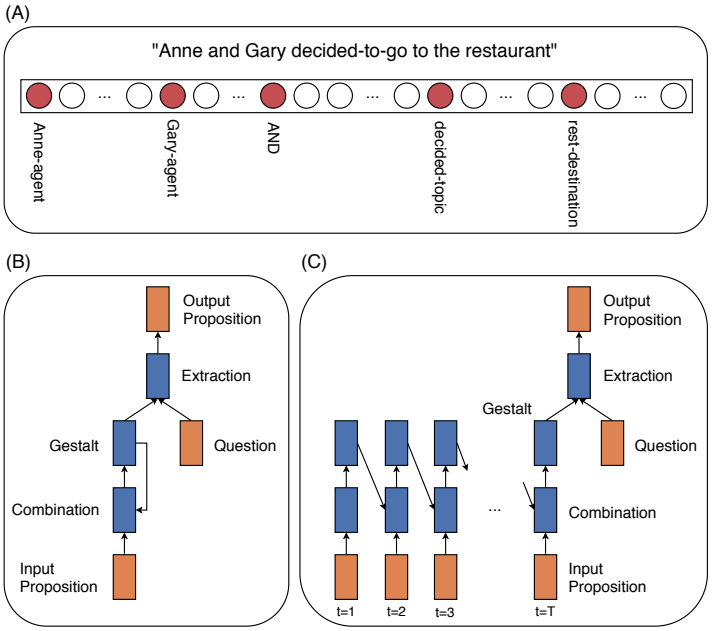

Figure 1. Story Gestalt model. (A) An example of a proposition as represented in the input layer. (B) Model architecture. (C) Model's operation unfolded over time. See text for details.

subsystems. The first "comprehension" subsystem (input proposition, combination and gestalt layers), receives each proposition of a story one at the time as input. The activation in the proposition layer feeds forward to the combination and gestalt layers (100 units each). The gestalt layer has recurrent connections to the combination layer, which allows the model to form a representation of the story presented so far (see Figure 1C). The second "query" subsystem (gestalt, question, extraction and output proposition layers), receives as input the activation of the gestalt layer and the question layer. The question layer (34 units) consists of a vector of units representing all topic concepts in a localist fashion. The extraction layer (100 units) combines the activation of the gestalt and question layers and feeds forward to the output layer, which has the same dimensionality as the input layer.

To train a single story the model is presented with increasing longer sequences of the story propositions and, after each successive sequence, is asked about the last proposition. For example, imagine a story composed by the last three propositions of the Restaurant story in Table 2 (i.e., "Anne paid bill", "Anne tipped waiter big", "waiter gave change to Anne"). This story would be trained by presenting the model with the sequences: ["Anne paid bill"], ["Anne paid bill", "Anne tipped waiter big"] and ["Anne paid bill", "Anne tipped waiter big", "waiter gave change to Anne"]. The question for each sequence would be the topic concept of the last proposition of the sequence (i.e., "paid", "tipped" and "gave") and the target (i.e., what the model was trained to output) would be the last proposition of each sequence (i.e., "Anne paid bill", "Anne tipped waiter big", "waiter gave change to Anne"). The difference between the actual output and the target is used to train the model through a standard gradient descend algorithm. Once trained, the model can recover the full proposition associated with each topic of a story. For example, if a trained SG model is presented with the complete sequence of sentences on Table 2 and then asked about the topic "decided" (by activating the corresponding localist unit in the question layer) the model would

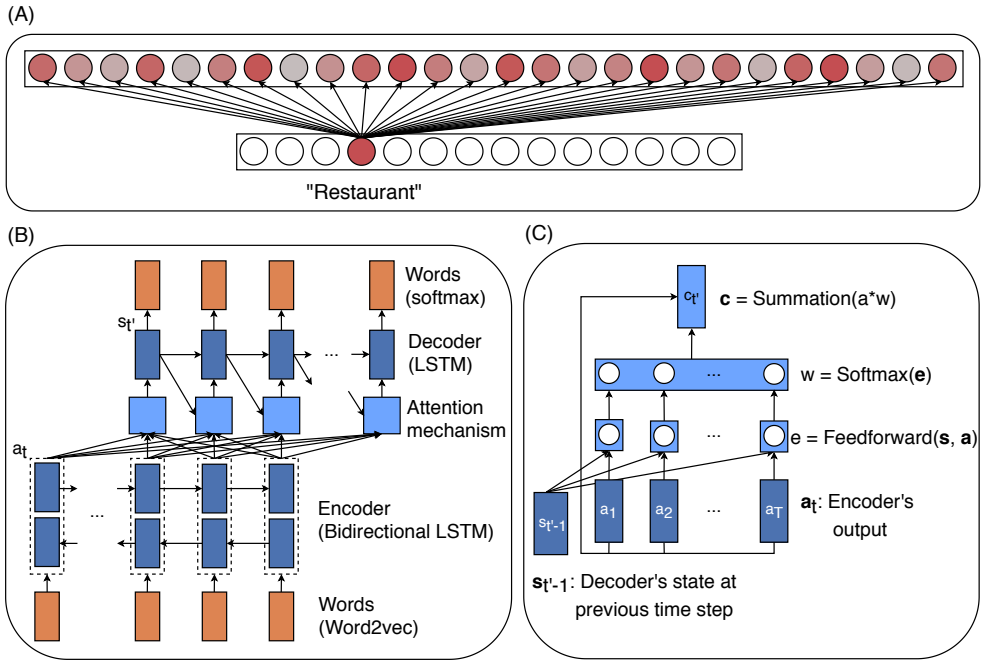

Figure 2. Seq2seq model with attention. (A) Input representation. (B) Model's architecture unfolded over time. (C) Attention mechanism. See text for details.

output an activation vector corresponding to the proposition "Anne and Gary decided to go to the restaurant".

St. John (1992) showed that the SG model can recover missing sentences from a story, review its predictions as it encounters new propositions and resolve pronouns. For example, if the model is presented with the complete sequence of propositions on Table 2 except for the third ("distance to restaurant far") and is asked about the topic "distance", the model would output an activation vector corresponding to the proposition "distance to restaurant far" because in its training data expensive restaurants are always far away (see Table 1).

2.2. Sequence-to-Sequence with Attention model

In order to test the performance of a contemporary deep learning system on our task, we implemented a version of the Seq2seq+Attention model (Bahdanau et al., 2015, see Figure 2)—a deep neural network architecture designed originally to solve machine translation problems. In translation problems, a source sentence in a given language (e.g., English) has to be translated into a different language (e.g., French). Typically, the source and target sentences have different lengths. In general, a Seq2seq model consist of an encoder network and a decoder network. Both are recurrent neural networks with their own independent time steps (t for the encoder and t' for the decoder in Figure 2B). The encoder transforms the input sequence into a sequence of fixed-size vectors and the decoder processes these transformed vectors to get the output sequence. Two important features of this model are the use of Word2Vec representations for the input words (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013) and an attention mechanism that allows the model to selectively "attend" to different parts of the encoder's output (Bahdanau et al., 2015).

Word2Vec embeddings (Mikolov et al., 2013) are dense distributed representations obtained by extracting the activation vector of the hidden layer of shallow neural network trained to predict the surrounding words given an input word in large corpus of text. Word2Vec representations maintain the distributional patterns of similarity between words, such that words used in similar contexts have similar representations (however, see Nematzadeh, Meylan, & Griffiths, 2017, for evidence of discrepancies between the patterns of similarity between Word2Vec representations and the patterns of similarity in human word association data). Our version of the Seq2seq+Attention model represents a single word at each time step t through a layer with localist units for each unique word in the data set (105 units). To represent a word the corresponding unit is given an activation of 1 while all other units are given an activation of 0 (i.e., a one-hot vector). This one-hot vector is transformed into a Word2Vec embedding (size 300) by a single feed-forward layer with a fixed set of weights (see Figure 2A). We did not allow the training process to change these weights.

The encoder (bottom part of Figure 2B) corresponds to a bidirectional long short-term memory neural network (Bidirectional LSTM, Graves & Schmidhuber, 2005). The Bidirectional LSTM is composed of two LSTM neural networks (250 units each in our model). The first LSTM reads the input from the beginning until the end of the sequence while the second reads the sequence in a backwards fashion. At each time step t both LSTMs produce their own output. The full output of the encoder at is the concatenation of the outputs of the forward and backward LSTMs. The encoder's output at each time step t can be understood as a summary of all precedent and following words to the current word with an emphasis on the words surrounding it (Bahdanau et al., 2015).

The attention mechanism (center part of Figure 2B and Figure 2C) corresponds to a feed-forward neural network that, at each decoder's time step t', takes as input the decoder previous state $s_{t'-1}$, and all encoder outputs a_1 to a_T (see Figure 2C). This feed-forward network produces a single number e_t for each encoder's output. This number is intended to capture the degree of alignment between the current word in the decoder with each word in the input sequence. This alignment score is normalized using a softmax function, yielding a single attention weight w_t for each encoder's output. The output of the attention mechanism is a context vector c'_t, which corresponds to the summation of all encoder's outputs weighted by their corresponding attention weight. In short, the vector c'_t represents a summary of the input words with an emphasis on the words that "correspond" better with the current output word.

The decoder (top part of Figure 2B) corresponds to a standard LSTM network (200 units) followed by feed-forward layer with softmax activation. This layer has a unit for each unique word in the data set (105 units) so that the decoder's output at each time step corresponds to a probability distribution over the dataset vocabulary. The model's answer at each time step is taken to be the word with maximum predicted probability. As this model is designed to receive words as inputs, during training we feed the propositions of our task to the model one word at the time. For each unfilled role we presented the special <NONE> word. After presenting the complete story, we input a special word <Q> to demarcate the beginning of the question, then we input the topic question, and finally we input a special word <GO> to tell the model to start the decoding process. The target output was the sequence of words corresponding to the full proposition involving the topic concept. The difference between the actual output and the target was used to train the model in the same way as in the SG model. Figure 3 presents an example of this process. Here, the Seq2seqs+Attention model (represented by the box) is presented with the complete sequence of words

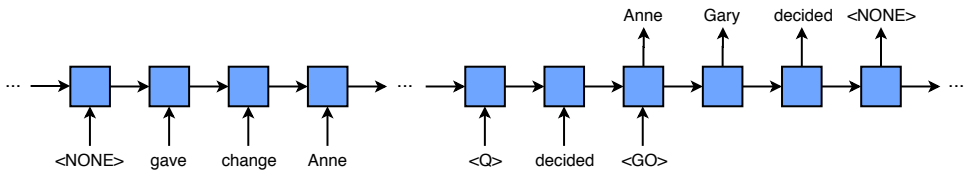

Figure 3. Training Process for the Seq2se2+Attention Model. See text for details.

Table 3. Example of a Concept Violation Story (Restaurant). Lois and Albert were restricted from instances of the Restaurant script during training.

Story	**Questions**	**Criteria**
1. <**Lois**> and <**Albert**> decided restaurant	decided	<**Lois**> and <**Albert**> decided restaurant
2. Restaurant quality <expensive>		
3. Distance to restaurant <far>		
4. <**Lois**> ordered <cheap-wine>	ordered	<**Lois**> ordered <cheap-wine>
5. <**Lois**> paid bill	paid	<**Lois**> paid bill
6. <**Lois**> tipped waiter <big>	tipped	<**Lois**> tipped waiter <big>
7. Waiter gave change to <**Lois**>	gave	Waiter gave change to <**Lois**>

corresponding to the Restaurant story in Table 2. The model is asked about the "decided" topic and it responds by outputting the sequence of words corresponding to the proposition "Anne and Gary decided to go to restaurant".

3. Simulation 1

In contrast to previous research with Gestalt models (Rabovsky et al., 2018; Rohde, 2002; St. John, 1992; St. John & McClelland, 1990), our manipulations aimed to disentangle the task's relational structure from its statistical structure. Specifically, our tests were designed to keep the relational structure of the test stories constant relative to the training data while varying their statistical properties. In short, these tests relied on capturing bindings between roles and fillers in specific instances while ignoring the statistical regularities from the training data. We termed our first test *concept violation*. In this test, we trained the models in the concept restricted condition and then tested them with stories where the agent-1, agent-2 or the patient-theme roles were filled by the restricted concepts. The questions consisted on all the topic concepts of the propositions in which the restricted concepts were used. A role-based answer to the question required using the restricted concept to fill the corresponding role. Table 3 presents an example of a Restaurant concept violation story. In this example, the concepts Albert and Lois had never appeared as agents in any Restaurant story during the model's training. The model was then tested using a story in which Albert or Lois appeared as agents in a Restaurant story by asking, for example, about the "tipped" proposition. The correct (role-based) answer was "Lois tipped waiter big". Note that, while the model was trained in stories where Lois appeared as an agent in other locations, and had been trained to output that someone tipped big with other agents, it had never been trained to output the exact proposition "Lois tipped waiter big". Table 3 also presents all the story questions and their corresponding role-based answers.

In our second test, termed *correlation violation*, we presented the models trained in the concept unrestricted condition with stories where we inverted a perfect statistical regularity of the story script. For example, in the Restaurant script the value of the

Table 4. Example of a Correlation Violation Story (Restaurant).

Story	Questions	Criteria
1. <Anne> and <Gary> decided restaurant 2. Restaurant quality <**expensive**> 3. Distance to restaurant <**near**> 4. <Anne> ordered <cheap-wine> 5. <Anne> paid bill 6. <Anne> tipped waiter <big> 7. Waiter gave change to <Anne>	distance	Distance to restaurant <**near**>

attribute role in the second proposition determines the value of the attribute role in the third proposition in that if the restaurant was cheap it was nearby and if it was expensive it was far away (see Table 1). To create a Restaurant correlation violation story, we switched the value of the attribute role in the third proposition (i.e., a cheap restaurant was now far away, and an expensive restaurant was now nearby). A role-based answer to the questions of this test would use the input concept in the third proposition to fill the attribute role, even though it corresponds to a violation of a correlation seen during training. Table 4 presents an example of a Restaurant correlation violation story, its question and corresponding role-based answer. In this example the model had been trained in Restaurant stories where expensive restaurants are always far away and cheap restaurants are always nearby and the model is tested in a Restaurant story where an expensive restaurant is close by. The model is asked about the "distance" proposition and the correct (role-based) answer is that the restaurant is close by (i.e., "Distance to restaurant near").

In our third test, termed *shuffled propositions*, we presented the models trained in the concept unrestricted condition with stories where we randomized the order of the propositions. Recall that in our stories there are no repeated topic concepts. As a direct consequence, a role-based answer to a question should use the concepts of the proposition corresponding to each question to fill its roles, ignoring the ordering of the propositions. Table 5 presents an example of a Restaurant shuffled propositions story, its questions and their corresponding role-based answers. In this example the model had been trained in stories that followed the same order of propositions as the Restaurant script (see Table 1). The model was presented with sequences of propositions that corresponded to a standard unrestricted Restaurant story, with the only difference being that the order of the propositions was randomized (e.g., the propositions in Table 5 are exactly the same as the ones on Table 2), so although the model had received all the individual propositions of the story during training, the model was never trained in the specific sequence being tested. After receiving the propositions, the model was asked about any of the topics of the story. For example, when asked about the "quality" topic, the correct (role-based) answer was the proposition "Restaurant quality expensive". It is worth to note that in all our tests the correct (role-based) answers required simply filling the roles of the answer proposition with the concepts that the model had received as input.

3.1. Training

We trained two versions of the SG model, one in 1,000,000 randomly generated concept restricted stories and another in 1,000,000 randomly generated concept unrestricted stories. We also trained two versions of the Seq2se2+Attention model, one in 500,000 randomly generated concept restricted stories and an-

Table 5. Example of a Shuffled Propositions Story (Restaurant).

Story	Questions	Criteria
4. <Anne> ordered <cheap-wine>	decided	<Anne> and <Gary> decided restaurant
5. <Anne> paid bill	quality	Restaurant quality <expensive>
1. <Anne> and <Gary> decided restaurant	distance	Distance to restaurant <far>
3. Distance to restaurant <far>	ordered	<Anne> ordered <cheap-wine>
7. Waiter gave change to <Anne>	paid	<Anne> paid bill
6. <Anne> tipped waiter <big>	tipped	<Anne> tipped waiter <big>
2. Restaurant quality <expensive>	gave	Waiter gave change to <Anne>

other in 500,000 randomly generated concept unrestricted stories. We used the Nadam optimization algorithm (Dozat, 2016) with default learning parameters. All our models were implemented in Keras (Chollet et al., 2015) with TensorFlow backend (Abadi et al., 2016). Full code for all simulations is available from `https://github.com/GuillermoPuebla/RelationReasonNN`.

3.2. Results

For each of our tests, we created a dataset of 728 randomly generated stories. This number corresponds to the number of all possible concept violation stories, which is the script with the lower number of possible stories. For all tests we compared the proposition generated by the model with the role-based answer. We coded the answer as correct (with a value of 1) if all the concept fillers in the answer corresponded to the concept fillers in the role-based answer and as a non-match (with a value of 0) otherwise. Figure 4 shows the proportion of correct answers per test and model. Recall that in our baseline test we presented the models trained in the concept unrestricted condition with concept unrestricted stories and asked questions about all the propositions in the stories (see Table 2 for an example of a Restaurant baseline story, its questions and corresponding correct answers). Because the test stories came from exactly the same distribution as the training stories this test is akin to a recall test of the training dataset. As can be appreciated in Figure 4, both models performed well in our baseline test. It is noteworthy that the Seq2seq+Attention model showed a better baseline performance than the SG model even though it was trained in half the number of stories (accuracy of 0.96 vs. 0.92).

Recall that in our concept violation test we trained the models in the concept restricted condition and then tested them with stories where the agent-1, agent-2 or the patient-theme roles were filled by the restricted concepts[1]. The questions consisted of all the topics of the propositions in which the restricted concepts were used and a correct (role-based) answer required using the restricted concepts to fill the corresponding roles (see Table 3 for an example of a Restaurant concept violation story, its questions and corresponding correct answers). In this test the SG model was unable to use the concepts restricted during training to answer the questions (accuracy of 0.08). Instead, the SG model almost invariably filled the roles of the restricted concepts with the most common concepts playing that role during training, which is a direct replication of the results of (St. John, 1992). For example, if the SG model was presented with a story like the one on Table 3 were the roles agent-1 and agent-2 corresponded to the restricted concepts "Lois" and "Albert", the model tended to output answers where the agent-1 and agent-2 were any of the other unrestricted agents (e.g., "Barbara" or "Clement"). The Seq2seq+Attention model performed significantly better at this test, achieving a slightly better level of accuracy than in the baseline test (accuracy of 0.99).

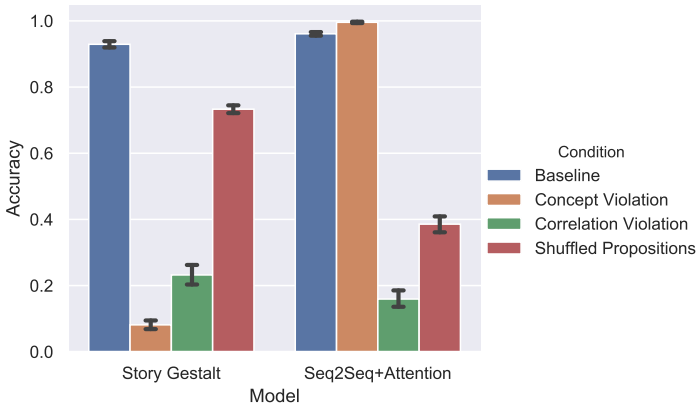

Figure 4. Accuracy per test and model. Both models perform well in the baseline condition. Furthermore, their performance was affected differentially in our critical conditions. The Story Gestalt model was more susceptible to the concept violation and correlation violation manipulations while the Seq2seq+Attention model was more susceptible to the correlation violation and shuffled propositions manipulations. As none of these manipulations changed the relational structure of the task, these results suggest that neither model was able to capture it during training. Error bars are 95% confidence intervals.

The attention mechanism seems to allow this model to apply its word representations to sequences where the words appeared in previously unseen stories.

Recall that in our correlation violation test we presented the models trained in the concept unrestricted condition with stories where we inverted a perfect statistical regularity of the story script and asked about the proposition that violated the perfect statistical regularity. The correct (role-based) answer required using the input concept even though it violated a statistical correlation from the training dataset. For example, because in the Restaurant script expensive restaurants are always far away, a Restaurant correlation violation story stated that an expensive restaurant is close by and the correct answer to the "distance" question was that the restaurant is indeed close by (see Table 4). Importantly, both models performed poorly in the correlation violation test, in other words, neither model was consistently able to correctly process texts that violated a perfect correlation seen in the training dataset (accuracy of 0.165 and 0.23 for the SG and Seq2seq+Attention models, respectively). Such behavior would seem quite unnatural for a human reader as it would be analogous to say that my friend John, who I just saw eating salad at the restaurant, ate chicken just because I've only seen him eating chicken in the restaurant in the past. Of course, it is possible to achieve perfect performance in this test by training the models in a corpus where all possible role-filler combinations appear in several contexts (e.g., several "establishments" other than the restaurant that are cheap and far away, cheap and close by, expensive and far away and expensive and close by, see e.g., St. John, 1992)[2]. However, the point of the simulation is that it shows that the inferences these models can make are in strictly limited by the statistical structure of its training corpus. It is noteworthy that the SG model achieved a higher accuracy than Seq2seq+Attention model in this test (although both models performed quite poorly). We suspect that the more powefull Seq2seq+Attention model is more likely to overfit to a perfect correlation in the dataset.

Recall that in our shuffled propositions test we presented the models trained in the concept unrestricted condition with concept unrestricted stories where the order of the propositions was randomized. A correct (role-based) answer required to use the con-

cepts of the proposition corresponding to each question to fill its roles, ignoring their ordering (see Table 5 for an example). While the randomization of the order of the propositions affected both models, the SG model performed significantly better than the Seq2seq+Attention model in this test (accuracy of 0.73 vs. 0.39). We hypothesize that the attention mechanism is the main reason for this difference in performance. Unfortunately, because of the length of our stories, taking out the attention mechanism yields the Seq2seq+Attention model unable to pass our baseline test (baseline performance around 0.5), so for now we were not able to test our hypothesis directly.

4. Simulation 2

Simulation 1 showed how a series of manipulations that should not affect a model that learns a relational representation of a story affects a classic and contemporary neural network model of language processing. This entails that neither model is learning a relation-based representation of the story, but instead they are relying on the statistical regularities of the training dataset to answer the questions. A potential issue with Simulation 1 is that the training objective of the task is rather indirect: it demands to learn to find the sentence the probe corresponds to from the test story. Arguably, this does not necessarily require to learn relationships between the objects and roles in the story to succeed at training time(although humans seem to naturally do so in equivalent situations Lake, Linzen, & Baroni, 2019).

To address this potential issue we adapted the original task of St. John (1992) to probe for relational roles directly. To accomplish this we added five new words to the models' vocabulary: *agent-1*, *agent-2*, *attribute*, *manner* and *patient*. In the Story Gestalt model these words corresponded to new localist units in the question layer and in the Seq2Seq+Attention model these words were added to the Word2Vec embeddings (we used the embeddings of the words *agent* and *actor* for *agent-1* and *agent-2*, respectively). We trained both models by presenting stories and asking about a specific role in the story. The models had to answer with the concept word that played that role in the story (see Table 6). In the SG model this mean to activate only the corresponding concept unit in the answer layer (as opposed to activate a group of units representing a sentence), while in the Seq2Seq+Attention model this meant to return a single concept word. Because in our stories the roles are specified at sentence level only a few roles remain constant in each story. In particular, the roles agent-1 and agent-2 are always filled by a single character throughout a story. This means that it is possible to test for relational generalization in the models by training these roles in a set of characters and test in a disjoint set. Importantly, these characters are seen during training across all scripts, just not filling the agent-1 and agent-2 roles. For this manipulation we created 4 new characters. The characters Will and Tina never filled the agent-1 role but were free to fill the agent-2 role and the characters Alex and Kate followed the opposite pattern.

Additionally, we sought to measure the models' answers to direct relational questions when there was a strong distribution shift at test time with known concepts like in the correlation violation condition of Simulation 1. For this we trained the models to answer direct relational questions to the role involved in the correlation violation manipulation while maintaining the same statistical regularities of Simulation 1. For example, in the restaurant stories we trained the models to answer a question about the attribute role in the third proposition. The models had to answer with the concept that filled that role (i.e., whether the restaurant was "near" or "far" which was

Table 6. Example of a Relational Probe Story (Restaurant).

Story	Questions	Criteria
1. <Anne> and <Gary> decided restaurant	agent-1	<Anne>
2. Restaurant quality <expensive>	agent-2	<Gary>
3. Distance to restaurant <**near**>	attribute	<**near**>
4. <Anne> ordered <cheap-wine>		
5. <Anne> paid bill		
6. <Anne> tipped waiter <big>		
7. Waiter gave change to <Anne>		

perfectly predictable from the quality of the restaurant, see Table 1). At test time the models were tested in a story where the filler of the role breaks the perfect correlation in the training distribution. For instance, the models were asked about the attribute role in the third proposition in a story where the restaurant was close by but it was expensive instead of cheap (see Table 6). Because the correlation violation manipulation is necessarily script-type specific, so it is the specific role asked about for each story type (see Appendix B for all the deterministic rules used in the correlation manipulation for each script type).

4.1. Training

We trained the models on randomly generated batches. A a single batch contained three story-question pairs, where the story across pairs was the same. The first question asked about the agent-1 role, the second about the agent-2 role and the third about the script-type-specific role. We trained the SG model in 200,000 batches and the Seq2Seq+Attention model in 30,000. Both models achieved ceiling performance during training. We used the same optimization algorithm and training parameters as in Simulation 1.

4.2. Results

We tested both models on 536 batches of stories where the filers of the roles corresponded to the usual fillers seen during training (baseline condition) and on 536 batches where the filers of the roles corresponded to the role-filler combinations withheld during training (relational condition). As can be appreciated in Figure 5, both models achieved good performance when the filers of the roles corresponded to the usual fillers seen during training. For example, both models would answer correctly to a question about the role agent-1 when Kate played that role (accuracy of 1.0 in the baseline condition for the agent-1 role). It is worth noting, however, that the Seq2Seq+Attention model performed slightly worse than the SG model in the baseline condition for the agent-2 role (accuracy of 0.74 vs. 1.0). It is also clear that both models performed worse when the agent-1 and agent-2 roles were filled by concepts that did not play those roles during training. In this case the Seq2Seq+Attention model performs slightly better than the SG model (accuracy of 0.2 vs. 0.0 for the agent-1 role and 0.52 vs. 0.18 for the agent-2 role).

Regarding the script-type-specific role, our results show that both models perform well in the baseline condition (accuracy of 0.96 for both models). In contrast, the Seq2Seq+Attention model performs significantly worse than the SG model in the relational condition (accuracy of 0.02 vs. 0.73). Note that unlike the agent-1 and agent-2 roles there is not a sharp division in the set of fillers of the baseline and relational

Figure 5. Accuracy per condition, role and model. Both models perform reasonably well in the baseline condition for all roles. For both models there is a significant drop in accuracy when the agent-1 and agent-2 roles are filled with concepts different than those used in training (relational condition). For the correlation-violating filler the drop in accuracy is more pronounced for the Seq2Seq+Attention model although is still appreciable in the SG model. As in Simulation 2 we probed for relational roles directly, these results strengthen the conclusion that neither model has grasped the relational structure of the task. Error bars are 95% confidence intervals.

conditions of the script-type-specific role. Instead, the main challenge of this test is to answer with the correct filler even though there is a strong distribution shift in the test stories. The poor performance of the Seq2Seq+Attention model in this task suggest that its comparatively better performance to the SG model in the agent-1 and agent-2 roles does not come from a more relational representation of the stories per se. The fact that a classic connectionist model performed better than a modern deep neural network in this task highlights how different experimental manipulations have different (and sometimes surprising) effects on different architectures, which necessitates to perform several tests when trying to characterize the relational reasoning capabilities of different models.

Overall, the results of this simulation mimic those of Simulation 1. It is not the case that the structure of the task and the training objective on Simulation 1 was the main factor that lead to the non-relational solution found by the models, as in this simulation we showed that directly probing for relational roles does not seem to improve the relational generalization capabilities of either model.

5. General discussion

We tested the relational processing capabilities of the SG model and the Seq2seq+Attention model, a classic connectionist model of text comprehension and a contemporary language processing deep learning architecture, respectively. In Simulation 1 we varied the statistical properties of the test stories while keeping their relational structure intact. Our results show that both models are able to use the statistical regularities of the training data to learn to answer questions correctly for stories that came from the same distribution as the training corpus. More importantly, however, our simulations demonstrate that the performance of both models is severely affected when the statistical properties of the test stories differ from those in the training corpus. Because we kept the relational structure of the test stories intact, our results show clearly that these models are not using the relational information of the

stories to answer the questions, but instead they are relying on the statistical regularities of the training dataset. In Simulation 2 we showed that this is true even when the models are asked directly about relational roles. In addition, although the technical advances of Seq2seq+Attention model made it able to pass our concept violation test in Simulation 1 this performance did not transfer to the direct relational questions of Simulation 2Overall, neither model showed a better capability to deal with relational reasoning tasks, as both models performed worse than the other in some condition of our simulations.

It is worth noting that the Seq2Seq+Attention model has been highly influential in the machine reading comprehension (MRC) literature. Attention mechanisms are a key component of virtually all mayor deep learning MRC architectures (for a review see Zhang, Yang, Li, & Wang, 2019). This literature has also produced a set of databases to test the reading comprehension capabilities of these systems. For example, the popular bAbI dataset (Weston, Bordes, Chopra, & Mikolov, 2016) consist of 20 tasks aimed to test basic forms of logical understating, such as deduction, induction, compound co-reference and many more. The texts and questions are generated automatically from a simulation of characters moving around and manipulating objects in a simple environment. Another popular dataset, the Stanford Question Answering Dataset (SQuAD) (Rajpurkar, Zhang, Lopyrev, & Liang, 2016), consist of a large set of questions generated by humans on a collection of passages extracted from Wikipedia. The answer to each question is a section of text from the corresponding article. Interestingly, different researchers have shown that these kinds of datasets are less rigorous tests of reading comprehension than previously thought. For example, (Kaushik & Lipton, 2018) showed that deep learning architectures can perform surprisingly well in many MRC datasets (including bAbI) even without seeing either the input text or the question. Another example is Jia and Liang (2017), who showed that deep learning models trained on SQuAD are susceptible to adversarial attacks that add untrained sentences, that share words with the correct answer, to the test texts (Jia & Liang, 2017). Notably, ungrammatical distractor sentences have a stronger adversarial effect than grammatical ones, which suggest that these models are relying in a superficial strategy to solve the reading comprehension task. We believe that highly controlled experiments such as the ones performed in the present research are necessary to evaluate neural network models (deep or otherwise) of language processing. Fortunately, some MRC researchers seem to taking this direction (Dunietz et al., 2020).

Our results are highly consistent with the findings of Lake and Baroni (2018) and Loula, Baroni, and Lake (2018), who found that sequence-to-sequence models (with and without attention mechanism) failed at a command-to-action translation task that required composing the meaning of new commands formed by using known primitive concepts combined in ways unseen during training. Even in the minority of cases were their models showed behavior that seemed compositional, they did it in a very non-human way (e.g., in one test their best performing model could correctly produce the action sequences corresponding to the instructions "turn left", and "jump right and turn left twice", but not the one corresponding to "jump right and turn left"). Hupkes, Dankers, Mul, and Bruni (2019) showed comparable results in a artificial grammar learning task with a Seq2seq+Attention model, a Convolutional Seq2seq model and a Transformer model.

Truly compositional behavior requires independent representations of objects and roles that can be bound together dynamically (i.e., compositional representations require a solution to the binding problem). In particular, compositionality results when a system can recursively apply predicate representations over other predicate repre-

sentation (e.g., *loves*(John, *loves*(Mary, Richard)), for discussions see Fodor, 1975; Marcus, 2001; Martin & Doumas, 2019; Tenenbaum, Kemp, Griffiths, & Goodman, 2011). We have shown that traditional PDP models (including current deep learning models) do not, as instantiated, perform dynamic binding. As a consequence, these models systematically fail when a task requires violating well learned statistical associations. As such, while there are certainly instances wherein the representations that these models learn will produce the same results as compositional representations, the resulting representations are not truly compositional.

One of the most important evolutionary advantages of relational reasoning is the ability to base inferences on relational roles disregarding the content of their arguments. This capacity allows us to make relational generalizations to completely new inputs (Penn et al., 2008). As traditional neural networks can't, by definition, make use of untrained units to perform successfully in given a task (Marcus, 1998), these models rely on spanning the input space to achieve good generalization (see Doumas & Hummel, 2012). Word embeddings like Word2Vec (Mikolov et al., 2013, cf. Miikkulainen and Dyer (1991)) can be seen as a technique to deal with this phenomenon. Even though in our Seq2seq+Attention model some concepts were not trained in some contexts, the vector representation of all concepts of a certain type (e.g., agents like "Anne" and "Lois") had similar representations because they appear in similar contexts in the Word2Vec training dataset. Another strategy to deal with new concepts (or new combinations of concepts) involves directly spanning the input space so that there are no truly new inputs to the model. For example, it is standard practice in neural networks research to make random splits of the data to obtain the training and test datasets. When the data are instantiations of relational structures (as in our tasks) this makes very likely that most objects appear as the fillers of most relational roles in the training dataset, which transforms the relational generalization problem in a interpolation problem, where the correct answer corresponds to an intermediate answer between two known cases (see Lake & Baroni, 2018, for a demonstration of the effects of random versus systematic splits on the training/test datasets). It is for this reason that traditional PDP models (e.g., O'reilly & Busby, 2002) and contemporary deep learning models (e.g., Hill, Santoro, Barrett, Morcos, & Lillicrap, 2019) targeted to solve relational reasoning tasks rely on spanning the input space in order to achieve high levels of generalization. Importantly, none of these techniques are solutions to the deeper problem of generalizing to new concepts or new combination of concepts based on abstract relations.

However, all of the above is not to say that neural network models cannot, in principle, integrate operations that allow them to implement a truly symbolic dynamic binding system. For example, the symbolic-connectionist models SHRUTI (Shastri & Ajjanagadde, 1993), LISA (Hummel & Holyoak, 1997, 2003), and DORA (Doumas et al., 2008; Doumas & Martin, 2018), use time as a binding signal that allows for role-filler independence and dynamic binding.

Interestingly, there has been a resurgence of interest on the binding problem in the neural networks (Besold et al., 2017; Franklin, Norman, Ranganath, Zacks, & Gershman, 2019) and computational neuroscience literature (Fitz et al., 2019; Pina, Bodner, & Ermentrout, 2018). Moreover, relational learning and reasoning have become a core topic on deep learning research (Bahdanau et al., 2018; Battaglia et al., 2018; Greff, Srivastava, & Schmidhuber, 2015; Hill et al., 2019; Santoro et al., 2017) with some deep learning architectures starting to implement operations traditionally associated with symbolic processing such as a content-addressable memory (Graves et al., 2016; Santoro, Bartunov, Botvinick, Wierstra, & Lillicrap, 2016; Weston, Chopra, & Bordes,

2014). Whether these non-traditional neural network architectures are capable of relational reasoning remains an open question that we plan to address in future research. Our results suggest, however, that for a model to successfully account for all aspects of relational processing, it will need to implement a solution to the binding problem.

Finally, while we herein illustrate the limitations of traditional neural networks when facing relational reasoning tasks, we hope that the results will motivate cognitive scientists and machine learning researchers to tackle the problem of relational learning and reasoning by first tackling the problem of dynamic binding. In the domain of neural network models, doing so will most likely will require us to go beyond the architectural constrains of traditional neural networks.

Acknowledgements

The work of Guillermo Puebla was supported by the PhD Scholarship Program of CONICYT, Chile. We thank Hugh Rabagliati for his comments on earlier versions of the manuscript.

Notes

[1]Although these concepts were never used the in the context of each specific script, they were seen in the training dataset as a whole. By definition, the output of any traditional neural network to a completely new (unseen) concept depends on its initial weights. Given that these weights are initialized randomly, the behavior of a neural network regarding an unseen input will be essentially random (Marcus, 1998).

[2]We actually run that simulation and, unsurprisingly, obtained perfect "generalization".

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... others (2016). Tensorflow: A system for large-scale machine learning. In *12th usenix symposium on operating systems design and implementation* (pp. 265–283).

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *International conference on learning representations.*

Bahdanau, D., Murty, S., Noukhovitch, M., Nguyen, T. H., de Vries, H., & Courville, A. (2018). Systematic generalization: What is required and can it be learned? *arXiv preprint arXiv:1811.12889*.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., ... Pascanu, R. (2018). *Relational inductive biases, deep learning, and graph networks.*

Besold, T. R., d'Avila Garcez, A., Bader, S., Bowman, H., Domingos, P., Hitzler, P., ... Zaverucha, G. (2017). *Neural-symbolic learning and reasoning: A survey and interpretation.*

Biederman, I. (1987). Recognition-by-components: a theory of human image understanding. *Psychological review*, *94*(2), 115.

Chen, D., Lu, H., & Holyoak, K. J. (2017). Generative inferences based on learned relations. *Cognitive science*, *41*, 1062–1092.

Chollet, F., et al. (2015). *Keras.*

Christie, S., & Gentner, D. (2010). Where hypotheses come from: Learning new relations by structural alignment. *Journal of Cognition and Development*, *11*(3), 356–373.

Doumas, L. A., & Hummel, J. E. (2005). Approaches to modeling human mental representations: What works, what doesn't and why. *The Cambridge handbook of thinking and reasoning, ed. KJ Holyoak & RG Morrison*, 73–94.

Doumas, L. A., & Hummel, J. E. (2012). Computational models of higher cognition. *Oxford handbook of thinking and reasoning*, 52–66.

Doumas, L. A., & Hummel, J. E. (2013). Comparison and mapping facilitate relation discovery and predication. *PloS one*, *8*(6), e63889.

Doumas, L. A., Hummel, J. E., & Sandhofer, C. M. (2008). A theory of the discovery and predication of relational concepts. *Psychological review*, *115*(1), 1.

Doumas, L. A., & Martin, A. E. (2018). Learning structured representations from experience. *Psychology of Learning and Motivation*, *69*, 165–203.

Dozat, T. (2016). Incorporating nesterov momentum into adam. In *International conference on learning representations.*

Dunietz, J., Burnham, G., Bharadwaj, A., Chu-Carroll, J., Rambow, O., & Ferrucci, D. (2020). To test machine comprehension, start by defining comprehension. *arXiv preprint arXiv:2005.01525*.

Falkenhainer, B., Forbus, K. D., & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial intelligence*, *41*(1), 1–63.

Fitz, H., Uhlmann, M., van den Broek, D., Duarte, R., Hagoort, P., & Petersson, K. M. (2019). Neuronal memory for language processing. *bioRxiv*, 546325.

Fodor, J. A. (1975). *The language of thought* (Vol. 5). Harvard university press.

Forbus, K. D., Liang, C., & Rabkina, I. (2017). Representation and computation in cognitive models. *Topics in cognitive science*, *9*(3), 694–718.

Franklin, N., Norman, K. A., Ranganath, C., Zacks, J. M., & Gershman, S. J. (2019). Structured event memory: a neuro-symbolic model of event cognition. *BioRxiv*, 541607.

Gentner, D. (2016). Language as cognitive tool kit: How language supports relational thought. *American psychologist*, *71*(8), 650.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, *18*(5-6), 602–610.

Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., ... others (2016). Hybrid computing using a neural network with dynamic external memory. *Nature*, *538*(7626), 471.

Greff, K., Srivastava, R. K., & Schmidhuber, J. (2015). *Binding via reconstruction clustering.*

Halford, G. S., Wilson, W. H., & Phillips, S. (1998). Processing capacity defined by relational complexity: Implications for comparative, developmental, and cognitive psychology. *Behavioral and Brain Sciences*, *21*(6), 803–831.

Hill, F., Santoro, A., Barrett, D., Morcos, A., & Lillicrap, T. (2019). Learning to make analogies by contrasting abstract relational structure. In *International conference on learning representations.*

Holyoak, K. J. (2012). Analogy and relational reasoning. *The Oxford handbook of thinking and reasoning*, 234–259.

Hummel, J. E., & Holyoak, K. J. (1997). Distributed representations of structure: A theory of analogical access and mapping. *Psychological review*, *104*(3), 427.

Hummel, J. E., & Holyoak, K. J. (2003). A symbolic-connectionist theory of relational inference and generalization. *Psychological review*, *110*(2), 220.

Hupkes, D., Dankers, V., Mul, M., & Bruni, E. (2019). The compositionality of neural networks: integrating symbolism and connectionism. *arXiv preprint arXiv:1908.08351*.

Jia, R., & Liang, P. (2017). Adversarial examples for evaluating reading comprehension systems. In *Empirical methods in natural language processing.*

Kaushik, D., & Lipton, Z. C. (2018, October-November). How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 5010–5015). Brussels, Belgium: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/D18-1546`

Kollias, P., & McClelland, J. L. (2013). Context, cortex, and associations: A connectionist developmental approach to verbal analogies. *Frontiers in Psychology*, *4*, 857.

Lake, B. M., & Baroni, M. (2018). Generalization without systematicity: On the compositional

skills of sequence-to-sequence recurrent networks. In *Icml*.

Lake, B. M., Linzen, T., & Baroni, M. (2019). Human few-shot learning of compositional instructions. In A. K. Goel, C. M. Seifert, & C. Freksa (Eds.), *Proceedings of the 41st annual conference of the cognitive science society* (pp. 611–617). Montreal, QB: Cognitive Science Society.

Leech, R., Mareschal, D., & Cooper, R. P. (2008). Analogy as relational priming: A developmental and computational perspective on the origins of a complex cognitive skill. *Behavioral and Brain Sciences*, *31*(4), 357–378.

Loula, J., Baroni, M., & Lake, B. M. (2018). Rearranging the familiar: Testing compositional generalization in recurrent networks. *arXiv preprint arXiv:1807.07545*.

Lu, H., Chen, D., & Holyoak, K. J. (2012). Bayesian analogy with relational transformations. *Psychological review*, *119*(3), 617.

Lu, H., Wu, Y. N., & Holyoak, K. J. (2019). Emergence of analogy from relation learning. *Proceedings of the National Academy of Sciences*, *116*(10), 4176–4181.

Marcus, G. F. (1998). Rethinking eliminative connectionism. *Cognitive psychology*, *37*(3), 243–282.

Marcus, G. F. (2001). *The algebraic mind: Integrating connectionism and cognitive science*. MIT Press Cambridge, MA.

Martin, A. E., & Doumas, L. A. (2019). Tensors and compositionality in neural systems. *Philosophical Transactions of the Royal Society B: Biological sciences*.

Medin, D. L., Goldstone, R. L., & Gentner, D. (1993). Respects for similarity. *Psychological review*, *100*(2), 254.

Miikkulainen, R., & Dyer, M. G. (1991). Natural language processing with modular pdp networks and distributed lexicon. *Cognitive Science*, *15*(3), 343–399.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Nematzadeh, A., Meylan, S. C., & Griffiths, T. L. (2017). Evaluating vector-space models of word representation, or, the unreasonable effectiveness of counting words near other words. In *Cogsci*.

O'reilly, R. C., & Busby, R. S. (2002). Generalizable relational binding from coarse-coded distributed representations. In *Advances in neural information processing systems* (pp. 75–82).

Penn, D. C., Holyoak, K. J., & Povinelli, D. J. (2008). Darwin's mistake: Explaining the discontinuity between human and nonhuman minds. *Behavioral and Brain Sciences*, *31*(2), 109–130.

Pina, J. E., Bodner, M., & Ermentrout, B. (2018). Oscillations in working memory and neural binding: A mechanism for multiple memories and their interactions. *PLoS computational biology*, *14*(11), e1006517.

Rabovsky, M., Hansen, S. S., & McClelland, J. L. (2018). Modelling the n400 brain potential as change in a probabilistic representation of meaning. *Nature Human Behaviour*, *2*(9), 693.

Rabovsky, M., & McClelland, J. L. (2020). Quasi-compositional mapping from form to meaning: a neural network-based approach to capturing neural responses during human language comprehension. *Philosophical Transactions of the Royal Society B*, *375*(1791), 20190313.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). Squad: 100, 000+ questions for machine comprehension of text. In J. Su, X. Carreras, & K. Duh (Eds.), *Proceedings of the 2016 conference on empirical methods in natural language processing, EMNLP 2016, austin, texas, usa, november 1-4, 2016* (pp. 2383–2392). The Association for Computational Linguistics. Retrieved from `https://doi.org/10.18653/v1/d16-1264`

Rogers, T. T., & McClelland, J. L. (2008). Précis of semantic cognition: A parallel distributed processing approach. *Behavioral and Brain Sciences*, *31*(6), 689–714.

Rogers, T. T., & McClelland, J. L. (2014). Parallel distributed processing at 25: Further explorations in the microstructure of cognition. *Cognitive science*, *38*(6), 1024–1077.

Rohde, D. L. (2002). *A connectionist model of sentence comprehension and production* (Unpublished doctoral dissertation). School of Computer Science, Carnegie Mellon University.

Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., & Lillicrap, T. (2016). Meta-learning with memory-augmented neural networks. In *International conference on machine learning* (pp. 1842–1850).

Santoro, A., Raposo, D., Barrett, D. G., Malinowski, M., Pascanu, R., Battaglia, P., & Lillicrap, T. (2017). A simple neural network module for relational reasoning. In *Advances in neural information processing systems* (pp. 4967–4976).

Shastri, L., & Ajjanagadde, V. (1993). From simple associations to systematic reasoning: A connectionist representation of rules, variables and dynamic bindings using temporal synchrony. *Behavioral and brain sciences*, *16*(3), 417–451.

St. John, M. F. (1992). The story gestalt: A model of knowledge-intensive processes in text comprehension. *Cognitive Science*, *16*(2), 271–306.

St. John, M. F., & McClelland, J. L. (1990). Learning and applying contextual constraints in sentence comprehension. *Artificial intelligence*, *46*(1-2), 217–257.

Tenenbaum, J. B., Kemp, C., Griffiths, T. L., & Goodman, N. D. (2011). How to grow a mind: Statistics, structure, and abstraction. *science*, *331*(6022), 1279–1285.

Van der Velde, F., & De Kamps, M. (2006). Neural blackboard architectures of combinatorial structures in cognition. *Behavioral and Brain Sciences*, *29*(1), 37–70.

Weston, J., Bordes, A., Chopra, S., & Mikolov, T. (2016). Towards ai-complete question answering: A set of prerequisite toy tasks. In Y. Bengio & Y. LeCun (Eds.), *4th international conference on learning representations, ICLR 2016, san juan, puerto rico, may 2-4, 2016, conference track proceedings.* Retrieved from `http://arxiv.org/abs/1502.05698`

Weston, J., Chopra, S., & Bordes, A. (2014). Memory networks. *arXiv preprint arXiv:1410.3916*.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., ... Dean, J. (2016). *Google's neural machine translation system: Bridging the gap between human and machine translation.*

Yuan, A. (2017). Domain-general learning of neural network models to solve analogy tasks–a large-scale simulation. In G. Gunzelmann, A. Howes, T. Tenbrink, & E. Davelaar (Eds.), *Proceedings of the 39th annual conference of the cognitive science society* (pp. 2081–2086). Austin, TX: Cognitive Science Society.

Zhang, X., Yang, A., Li, S., & Wang, Y. (2019). Machine reading comprehension: a literature review. *arXiv preprint arXiv:1907.01686*.

Appendix A. Concepts

Table A1. Concepts Used in all the Scripts.

Roles	Concepts
agents	Albert, Clement, Gary, Adam, Andrew, Lois, Jolene, Anne, Roxanne, Barbara, he, she, jeep, station-wagon, Mercedes, Camaro, policeman, waiter, judge, AND
topics	decided, distance, entered, drove, proceeded, gave, parked, swam, surfed, spun, played, weather, returned, mood, found, met, quality, ate, paid, brought, counted, ordered, served, enjoyed, tipped, took, tripped, made, rubbed, ran, tired, won, threw, sky
patients or themes	Albert, Clement, Gary, Adam, Andrew, Lois, Jolene, Anne, Roxanne, Barbara, he, she, jeep, station-wagon, Mercedes, Camaro, ticket, volleyball, restaurant, food, bill, change, chardonnay, prosecco, credit-card, drink, pass, slap, cheek, kiss, lipstick, race, trophy, frisbee
recipients or destinations	Albert, Clement, Gary, Adam, Andrew, Lois, Jolene, Anne, Roxanne, Barbara, he, she, jeep, station-wagon, Mercedes, Camaro, beach, home, airport, gate, restaurant, waiter, park
locations	beach, airport, restaurant, bar, race, park
manners	long, short, fast, free, pay, big, small, not, politely, obnoxiously
attribute	far, near, sunny, happy, raining, sad, cheap, expensive, clear, cloudy

Appendix B. Story Scripts

Table B1. Park Script.

Script
<agent-1> and <agent-2> decided to go to the park The distance to the park was <near/far> <agent-1> got in <vehicle> <agent-1> drove <vehicle> to the park for a <short/long> time <agent-1> proceed to the park fast <agent-1> parked at the park for <free/pay> The weather was sunny <agent-1> ran through the park <He/She> threw a Frisbee to <agent-1/agent-2>
Concept restrictions
The roles agent-1 and agent-2 never correspond to 'Clement' or 'Roxanne'
Deterministic rule
The distance to the park determines driving time completely: $near \rightarrow short$, $far \rightarrow long$

Table B2. Bar Script.

Script

<agent-1> met <agent-2> at the bar
AND if agent1 = rich (1.0):
  <agent-1> enjoyed expensive-wine at the bar
AND if agent1 = cheap (1.0):
  <agent-1> did not enjoy expensive-wine at the bar
<agent-2> ordered a drink to the waiter at the bar
AND if agent2 = rich (1.0):
  The drink was expensive
AND if agent2 = cheap (1.0):
  The drink was cheap
OR (2):
  (0.5):
  <agent-2> made a polite pass at <agent-1>
  OR (2):
  (0.3):
  <agent-1> gave a slap to <agent-2>
  <agent-2> rubbed cheek
  (0.7):
  <agent-1> gave a kiss to <agent-2>
  <agent-2> rubbed lipstick
  (0.5):
  <agent-2> made a obnoxious pass at <agent-1>
  OR (2):
  (0.7):
  <agent-1> gave a slap to <agent-2>
  <agent-2> rubbed cheek
  (0.3):
  <agent-1> gave a kiss to <agent-2>
  <agent-2> rubbed lipstick

Concept restrictions
The roles agent-1 and agent-2 never correspond to 'Andrew' or 'Barbara'
Deterministic rule
The action of agent-1 determines what agent-2 rubes completely: $slap \rightarrow cheek$, $kiss \rightarrow lipstick$

Table B3. Airport Script.

Script
<agent-1> decided to go to airport Distance to airport <near/far> <agent-1> found change <agent-1> drove <vehicle> to airport <short/long> <agent-1> ran to gate <agent-1> met <agent-2> at airport <agent-1> <agent-2> returned home
Concept restrictions
The roles agent-1 and agent-2 never correspond to 'Gary' or 'Jolene'
Deterministic rule
The distance to the airport determines driving time completely: $near \rightarrow short$, $far \rightarrow long$

Table B4. Beach Script.

Script
<agent> decided to go to the beach The beach was far away OR (2): (0.5): <agent> entered <vehicle> <agent> drove <vehicle> to the beach for a long time AND if agent1 = male (1.0): <agent> proceeded <vehicle> to the beach fast AND (0.5): The policeman gave a ticket to <agent> (0.5): <agent> drove <vehicle> to the beach for a long time AND (0.8): <agent> swam in the beach <agent> won the race in the beach AND if agent1 = male (0.87): <agent> surfed on the beach <agent> spun AND if agent1 = female (0.33) <agent> surfed on the beach AND (0.33): <agent> played volleyball in the beach OR (2) (0.8) The weather was <sunny> <agent> returned home for a long time <agent> was in a <happy> mood (0.2): The weather was <cloudy> <agent> returned home for a long time <agent> was in a <sad> mood
Concept restriction
The roles recipient and patient never correspond to 'Camaro'
Deterministic rule
The weather determines the agent's mood completely: *sunny* → *happy*, *cloudy* → *sad*